IN THE CLAIMS:

Please amend the claims as follows.

1. (Currently Amended) A dirty memory subsystem for a computer system, the dirty memory subsystem comprising:

   storage operable to store redundant copies of dirty indicators, each dirty indicator being associated with a respective block of main memory and being settable to a predetermined state to indicate that the block of main memory associated therewith has been dirtied; and

   a control logic operable to read the redundant copies of a dirty indicator from storage and to treat the block of memory associated therewith as dirtied if all or any one of the redundant copies of the dirty indicator has the predetermined state.

2. (Cancelled)

3. (Original) The dirty memory subsystem of claim 2, wherein the control logic is operable to cause the block of memory associated with a dirty indicator for which at least one copy thereof has the predetermined state to be copied from the main memory to another memory.

4. (Original) The dirty memory subsystem of claim 3, wherein the other memory is another main memory.

5. (Original) The dirty memory subsystem of claim 1, operable to reset each of the redundant copies of a dirty indicator to a state other than the predetermined state after reading the redundant copies of the dirty indicator.

6. (Original) The dirty memory subsystem of claim 1, wherein the storage comprises at least two memory units each for storing a redundant set of dirty indicators.

7. (Original) The dirty memory subsystem of claim 6, wherein the control logic includes comparison logic for each of the memory units for determining whether a copy of the dirty indicator is set to the predetermined state.

8. (Original) The dirty memory subsystem of claim 7, wherein the control logic includes output logic responsive to the output of each comparison logic for determining whether a block of memory associated with a dirty indicator is to be treated as dirtied.

9. (Original) The dirty memory subsystem of claim 1, wherein each dirty indicator comprises a single bit.

10. (Original) The dirty memory subsystem of claim 1, wherein a block of main memory is a page of main memory.

11. (Original) The dirty memory subsystem of claim 1, comprising a hierarchical dirty memory.

12. (Currently Amended) A computer system comprising a dirty memory subsystem and at least one processing set that includes main memory, the dirty memory subsystem comprising:

storage operable to store redundant copies of dirty indicators, each dirty indicator being associated with a respective block of main memory and being settable to a predetermined state to indicate that the block of main memory associated therewith has been dirtied; and

a control logic operable to read the redundant copies of a dirty indicator from storage and to treat the block of memory associated therewith as dirtied if all or any one of the redundant copies of the dirty indicator has the predetermined state.

13. (Original) The computer system of claim 12, comprising a plurality of processing sets that each include main memory.

14. (Original) The computer system of claim 13, wherein the processing sets are operable in lockstep, the computer system comprising logic operable to attempt to reinstate an equivalent memory state in the main memory of each of the processor following a lockstep error.

15. (Currently Amended) A method of managing reinstatement of an equivalent memory state in the main memory of a plurality of processing sets of a fault tolerant computer following a lock step error, wherein a dirty memory subsystem stores redundant copies of dirty indicators that are settable to a predetermined state indicative that a block of main memory associated therewith has been dirtied, the method including the performance of at least one cycle of copying any page of main memory that has been dirtied from a first processing set to each other processing set, each cycle including reading the redundant copies of a dirty ~~indicators~~ indicator from storage and treating a block of memory associated with the redundant copies as dirtied if ~~at least~~ all or any one of the redundant copies of the dirty indicator has the predetermined state.

16. (Original) The method of claim 15, a block of memory associated with the dirty indicator for which at least one copy has the predetermined state is copied from the main memory to another memory.

17. (Original) The method of claim 16, wherein the other memory is another main memory.

18. (Original) The method of claim 15, wherein each of the redundant copies of the dirty indicator are reset to a state other than the predetermined state after reading the redundant copies of the dirty indicator.

19. (Original) The method of claim 15, comprising maintaining at least two copies of a dirty indicator in at least two memory units.

20. (Original) The method of claim 19, comprising separately assessing the state of copies of a dirty indicator for each of the memory units for determining whether a dirty indicator is set to the predetermined state.

21. (Original) The method of claim 20, comprising responding to each assessment and determining that the block of memory is dirty if any assessment indicates that the dirty indicator has the predetermined state.

22. (Original) The method of claim 15, wherein each dirty indicator comprises a single bit.

23. (Original) The method of claim 15, wherein a block of main memory is a page of main memory.

24. (Original) The method of claim 15, comprising a hierarchical dirty memory.

25. (New) The dirty memory subsystem of claim 1, wherein, after both redundant copies of the dirty indicator are read from storage, any difference between the stored redundant copies of the dirty indicator is considered as indicative of memory corruption.